МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

‘Функциональная схемотехника’

Вариант №6

Выполнил:
Студент группы P33312
Соболев Иван
Александрович
Преподаватель:
Табунщик Сергей
Михайлович

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2024

Цели работы:

1. Получить базовые знания о принципах построения цифровых интегральных схем с использованием технологии КМОП.

2. Познакомиться с технологией SPICE-моделирования схем на транзисторах.

3. Получить навыки описания схем базовых операционных элементов (БОЭ) комбинационного типа на вентильном уровне с использованием языка описания аппаратуры Verilog HDL.

Задание:

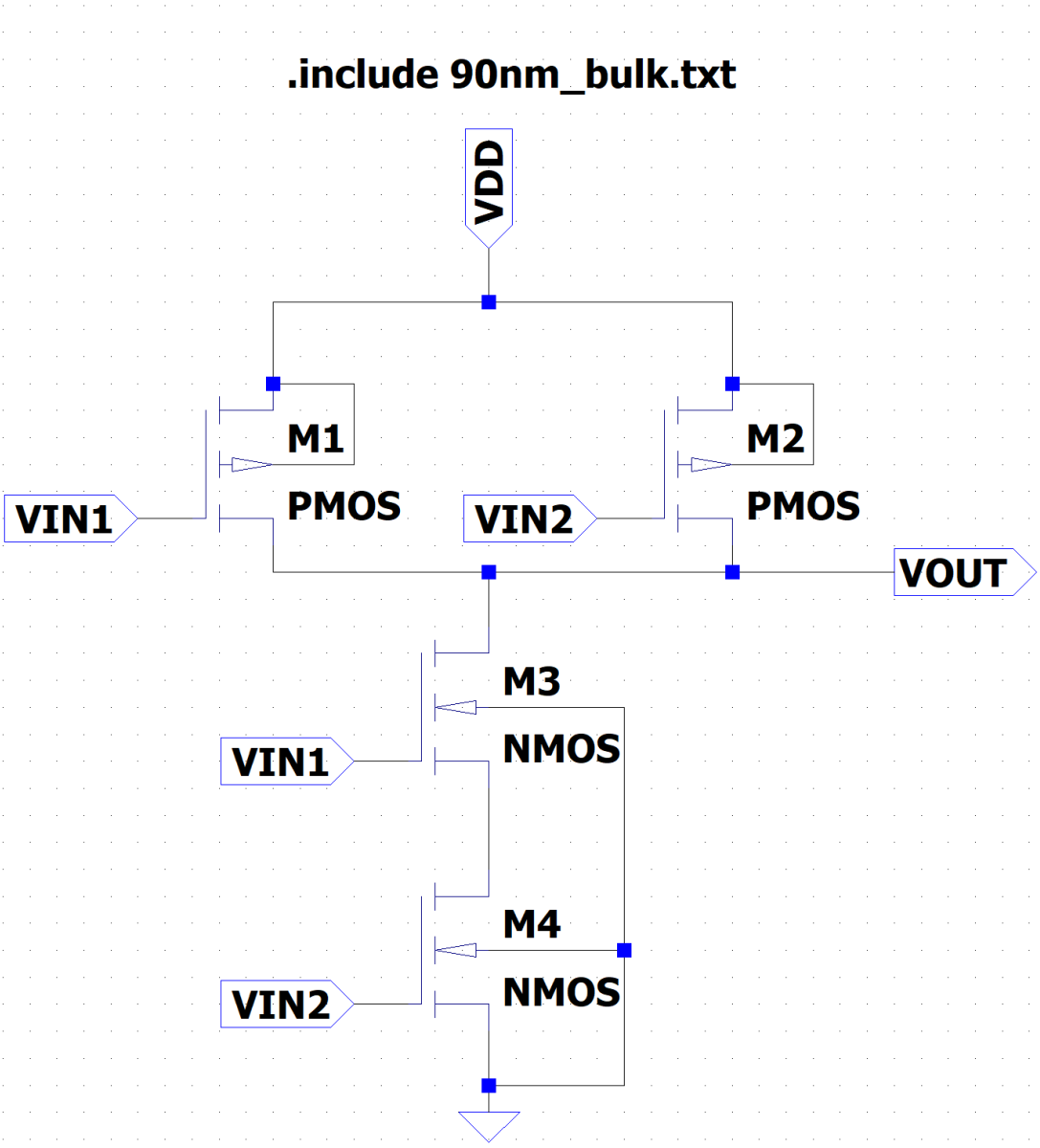
Логический базис – NAND; БОЭ – позиционный дешифратор «3 в 8».

Часть №1. LTSpice.

Разработка вентиля.

Схема разработанного вентиля:

VIN{1, 2} – входы, VOUT - выход, VDD – напряжение питания; использовано по 2 транзистора PMOS и NMOS.



Символ вентиля:

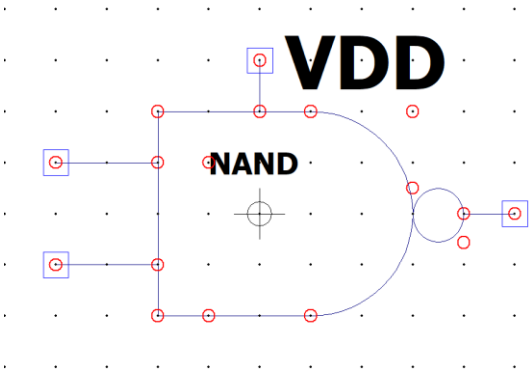
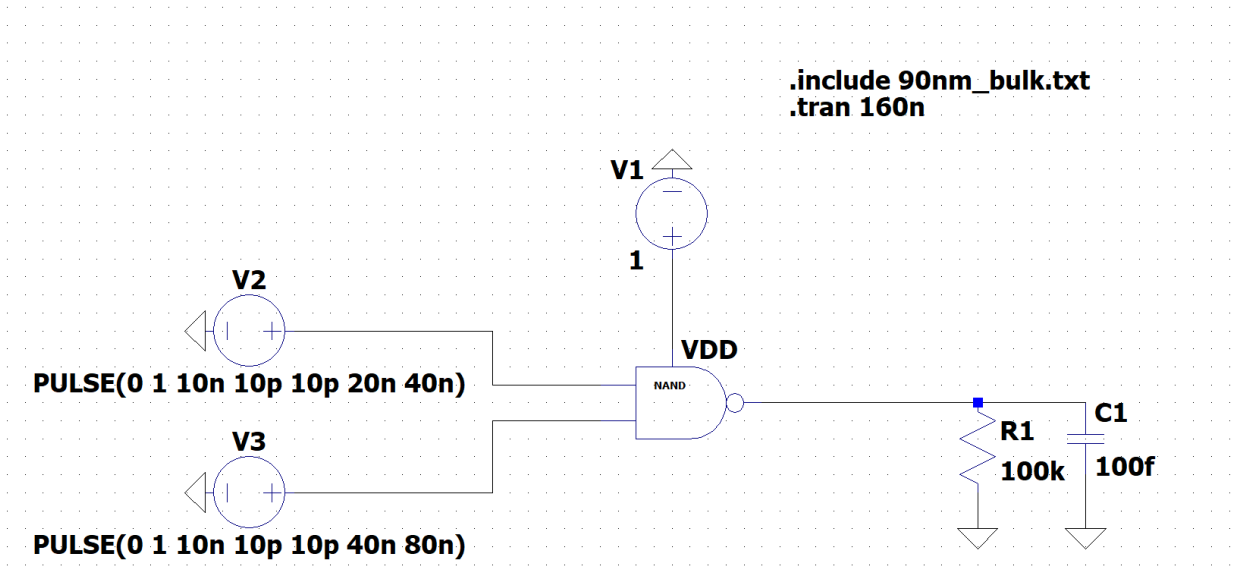
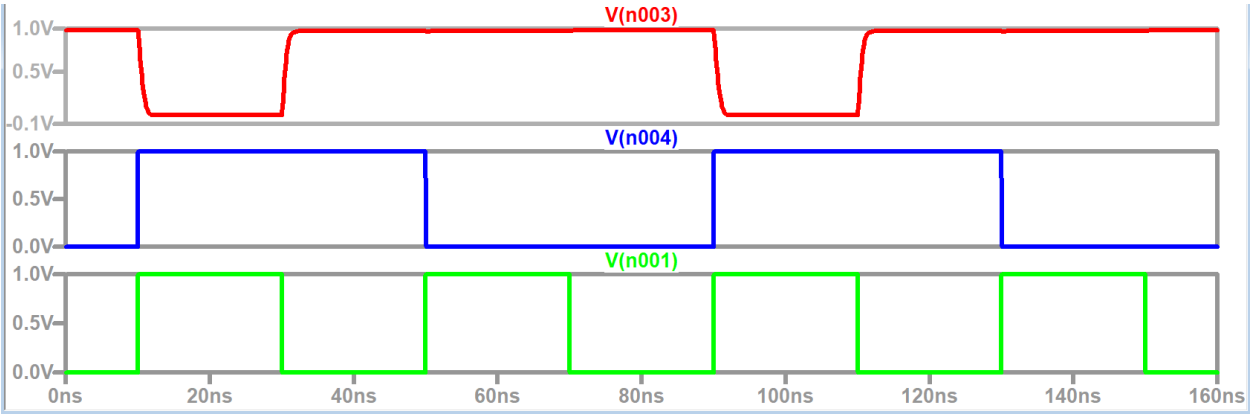


Схема тестирования:



VDD = VIN = 1 В; начальное напряжение - 0 В, активное напряжение - 1 В, задержка запуска - 10 нс, время фронта и спада - 10 пс, активное время первого источника напряжения - 20 нс, период - 40 нс, для следующего последние две характеристики в два раза больше, а частоты, соответственно, меньше; резистор и конденсатор отвечают за имитацию задержки.

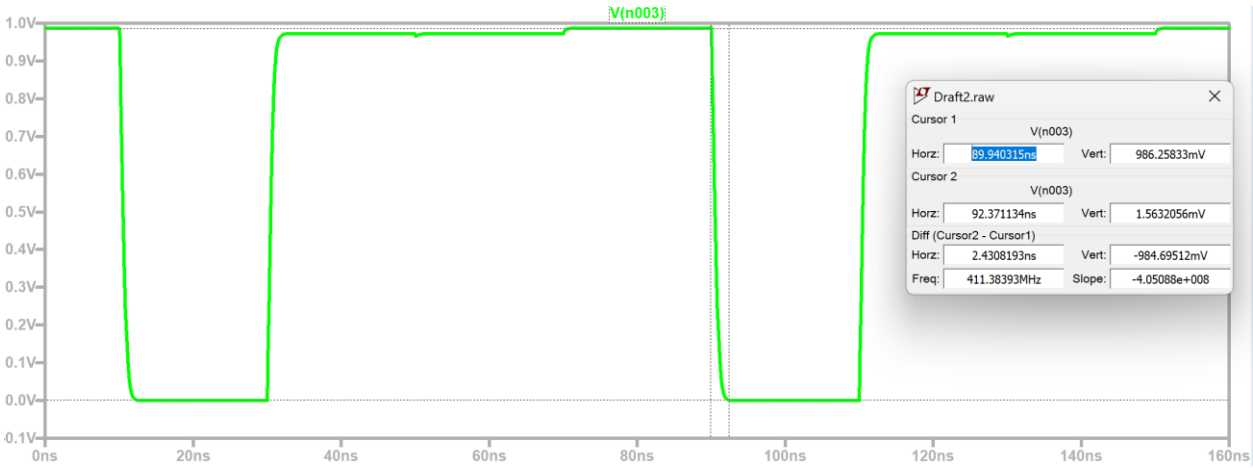
Временная диаграмма процесса тестирования вентиля:



1 В – логическая единица, 0 В – логический ноль; NAND подаёт на выход единицу во всех случаях, кроме равенства единице обоих входных сигналов.

Это отображено на рисунке – на 10 и 90 нс выходной сигнал падает в ноль и остаётся таким в течение 10 нс, после чего выходит обратно в единицу.

Результат измерения задержки распространения сигнала через вентиль:



Два курсора на верхней (~ 1 В) и нижней (~ 0 В) границах заднего фронта.

Задержка равна T ~ 2,4 нс

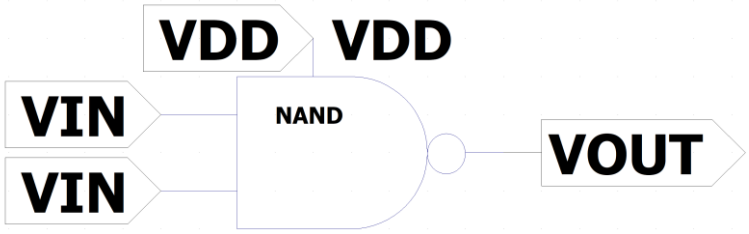
Тогда максимальная частота работы вентиля равна f = 1/T = 417 МГц

Разработка БОЭ.

На базе данного вентиля для удобства разработки БОЭ (дешифратора 3 в 8) я создал также инвертор и NAND с 4 входами. Инвертор необходим для конечного инвертирования. Так как обычно схема инвертора состоит из AND, а мы используем NAND – надо еще раз инвертировать.

Схема и символ инвертора.

Логическое выражение: $\underline{A}=\underline{A\&A}$



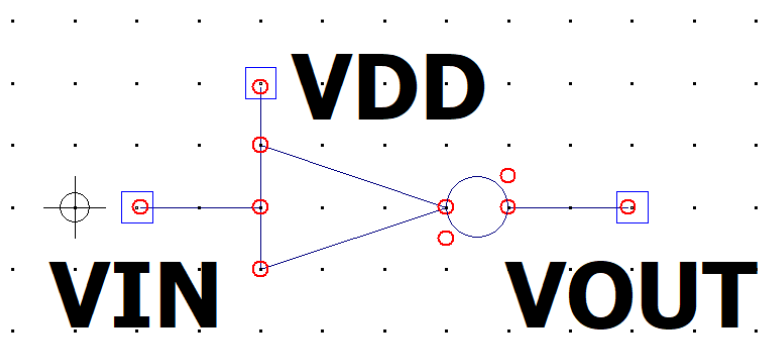
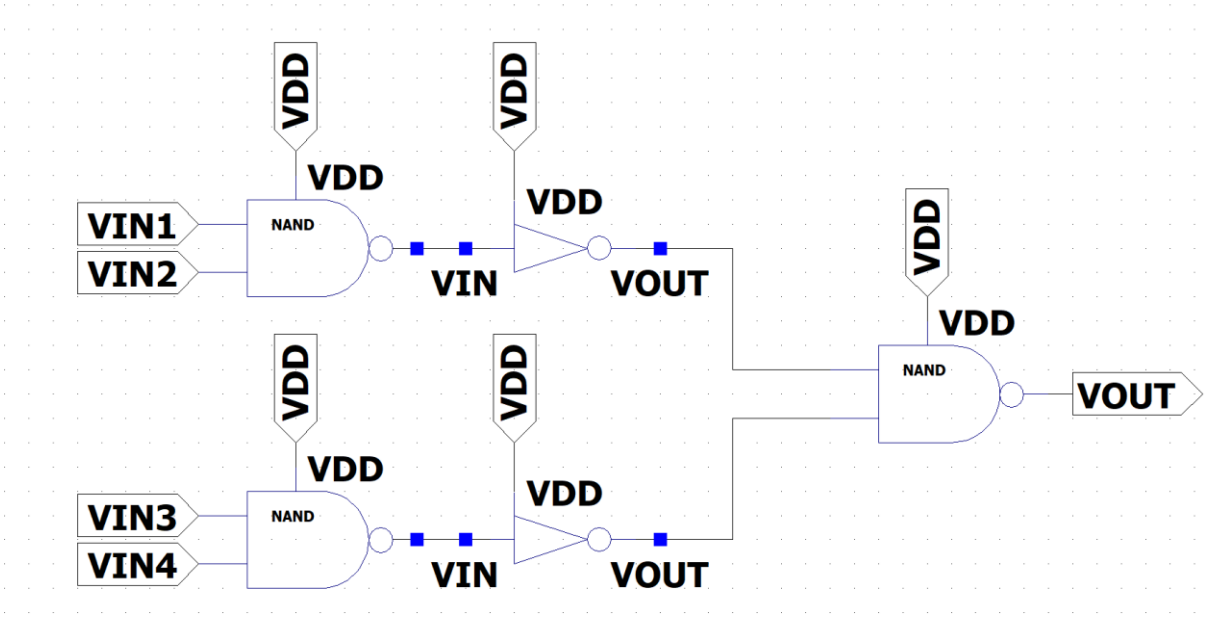
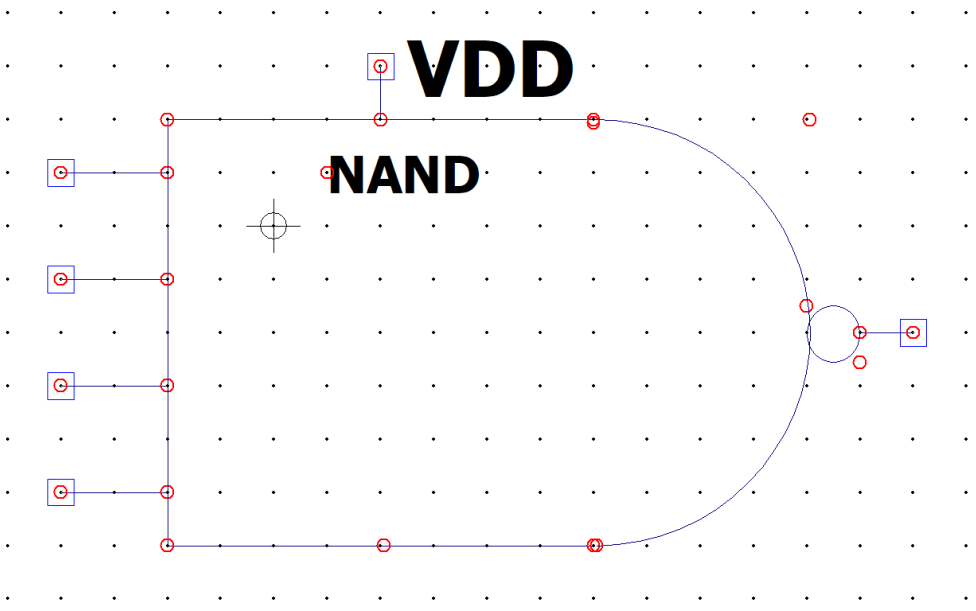


Схема и символ NAND с 4 входами.

Логическое выражение: $\underline{A\&B\&C\&D} = \underline{\underline{A\&B}}\&\underline{\underline{C\&D}}$





Временная диаграмма процесса тестирования вентиля:

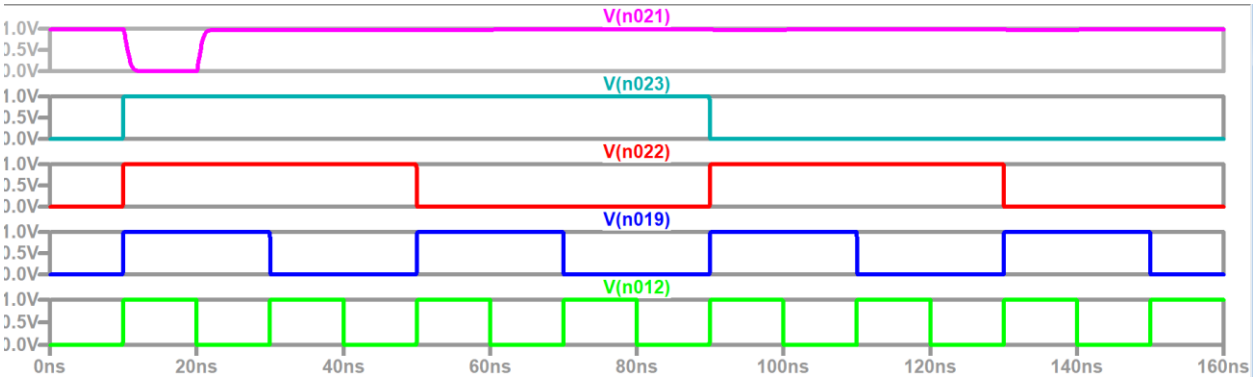
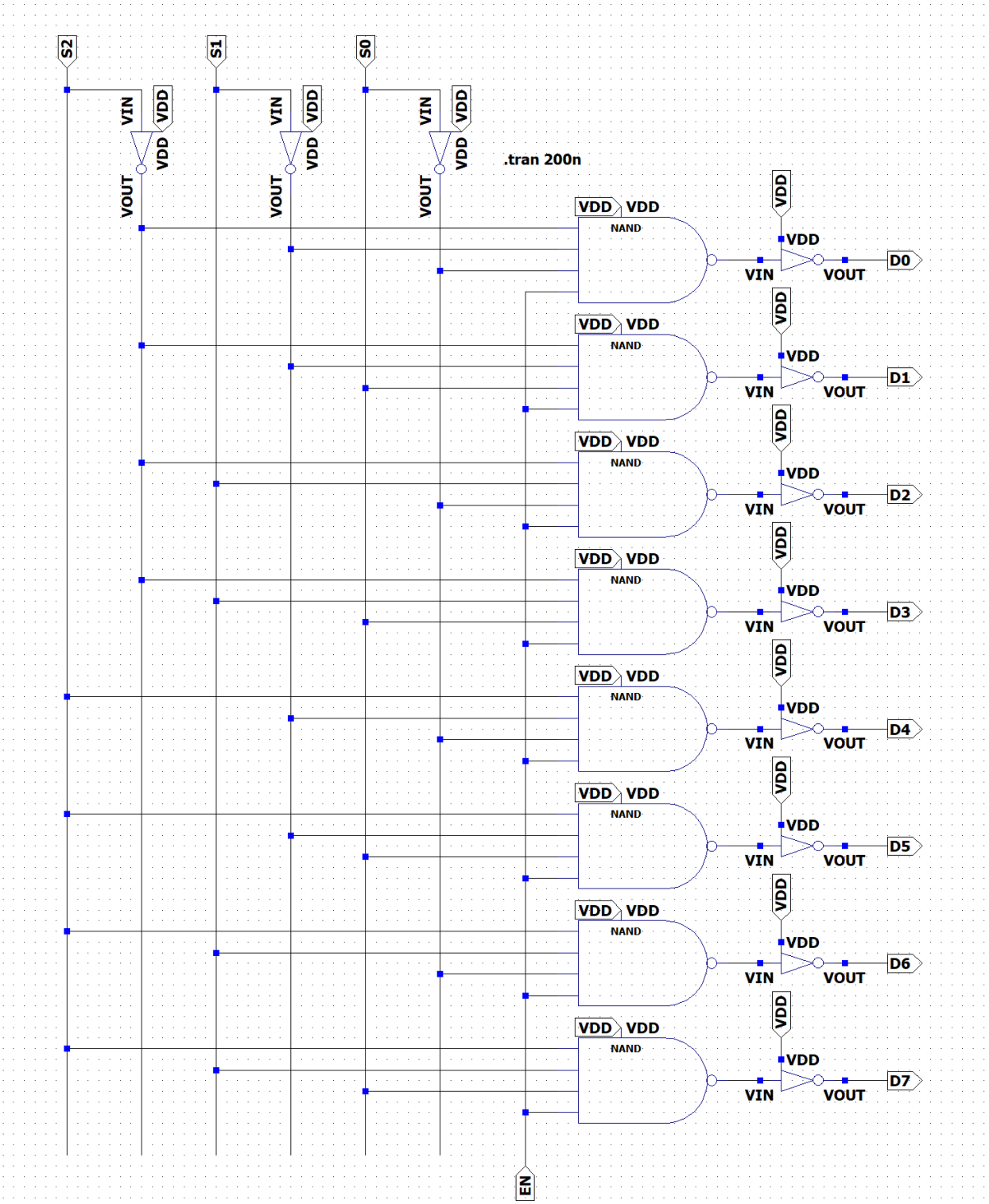


Схема разработанного БОЭ:



Модуль преобразует трёхразрядное двоичное число на входе в десятичное на выходе (вернее, в его унитарный код). На вход модулю подаётся двоичное число от 0 до 7 (входы S{0-2} соответствуют его разрядам, S2 - старший, S0 - младший), на выходе находятся 8 бит (D{0-7}), один из которых, порядок которого соответствует входу, становится равным 1, если сигнал разрешения EN (Enable), отвечающий за активность дешифратора, равен 1 (при равном 0 весь выход тоже будет нулевой). При изменении входа единице станет равен уже другой бит, а предыдущий обнулится.

Символ разработанного БОЭ:

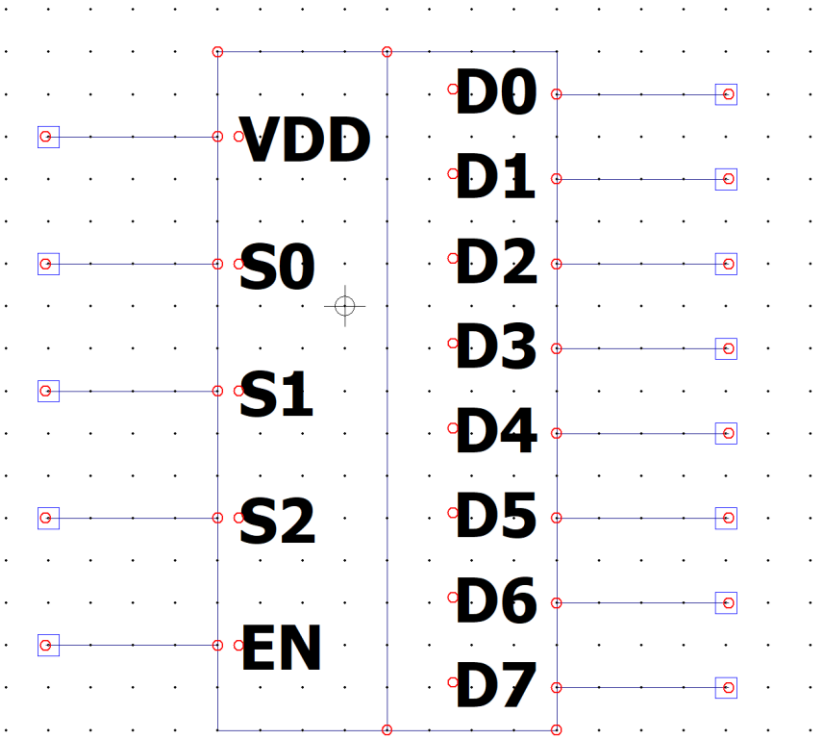
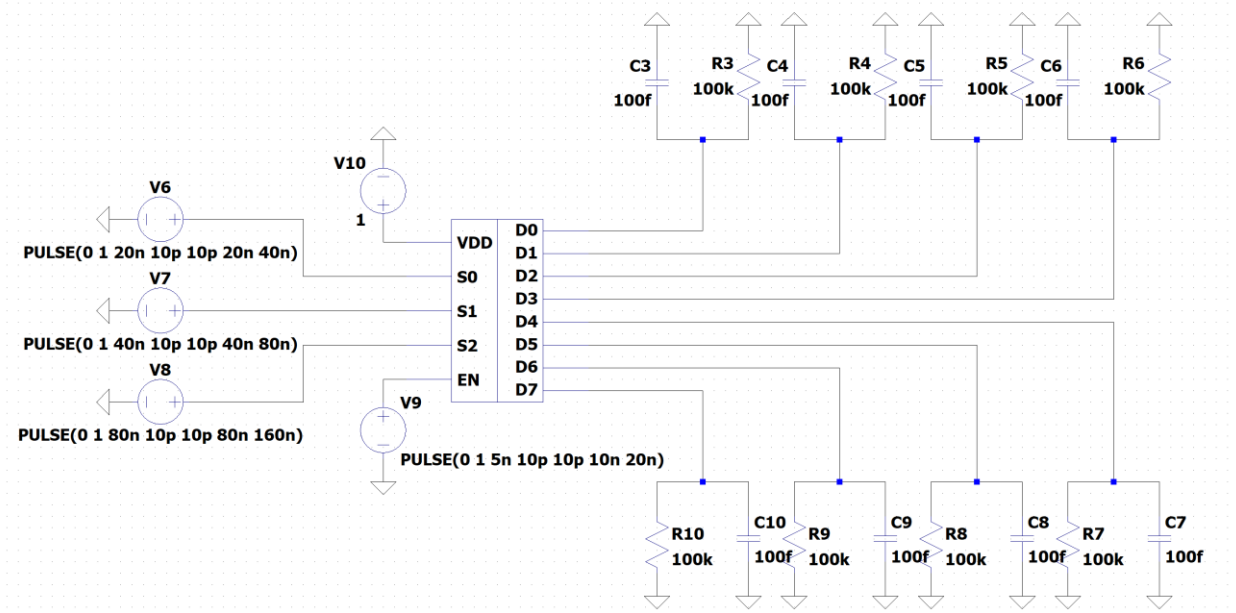
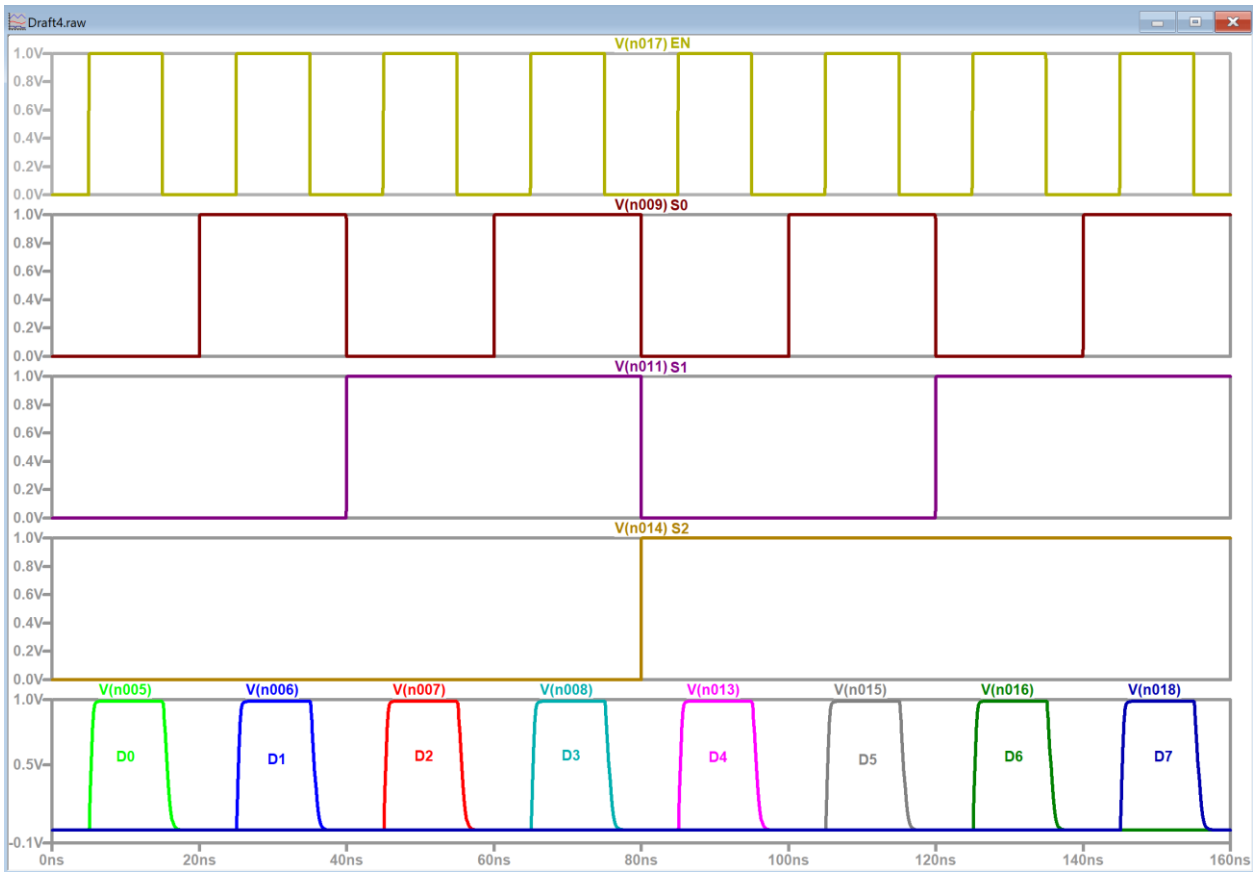


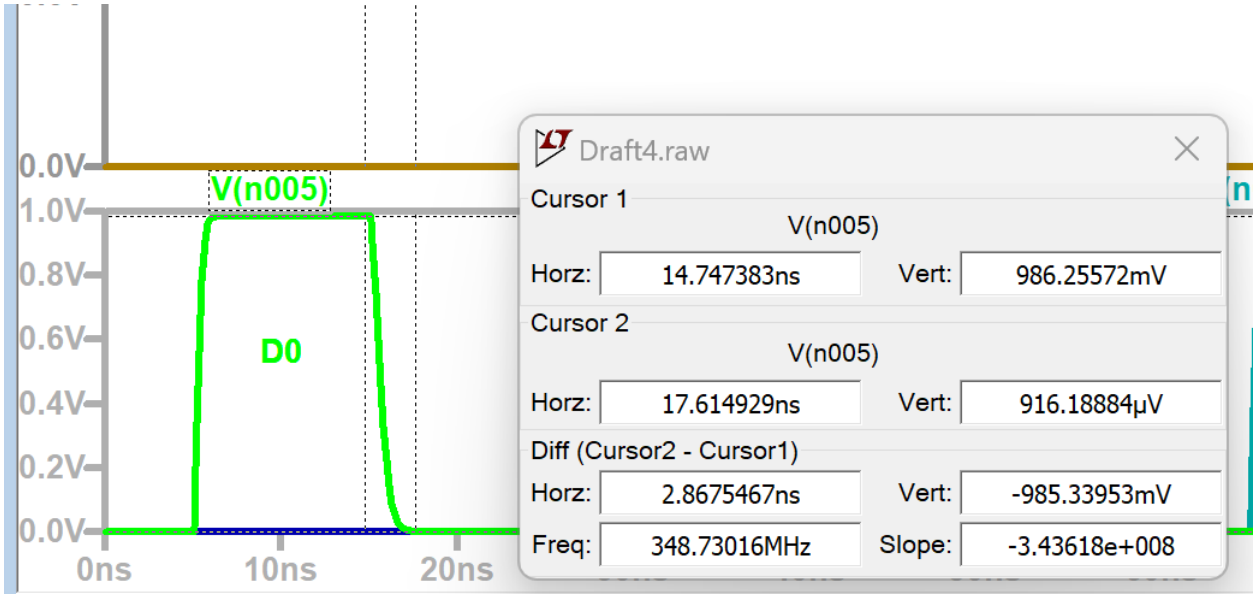
Схема тестирования:



Временная диаграмма процесса тестирования БОЭ:



Результат измерения задержки распространения сигнала через БОЭ:



Два курсора на верхней (~ 1 В) и нижней (~ 0 В) границах заднего фронта.

Задержка равна T ~ 2,87 нс

Тогда максимальная частота работы БОЭ равна f = 1/T = 348 МГц

Часть № 2. Verilog.

Код разработанного модуля БОЭ:

```
`timescale 1ns / 1ps


module decoder(
    input [2:0] s,
    input en,
    output [7:0] d
);


    wire [2:0] not_s;
    wire [7:0] not_d;
    wire [7:0] s_2_1;
    wire [7:0] not_s_2_1;
    wire [7:0] s_0_en;
    wire [7:0] not_s_0_en;


    nand(not_s[2], s[2], s[2]);
    nand(not_s[1], s[1], s[1]);
    nand(not_s[0], s[0], s[0]);


    nand(not_s_2_1[0], not_s[2], not_s[1]);
    nand(not_s_0_en[0], not_s[0], en);
    nand(s_2_1[0], not_s_2_1[0],not_s_2_1[0]);
    nand(s_0_en[0], not_s_0_en[0], not_s_0_en[0]);
```

```
nand(not_d[0], s_2_1[0], s_0_en[0]);

nand(not_s_2_1[1], not_s[2], not_s[1]);
nand(not_s_0_en[1], s[0], en);
nand(s_2_1[1], not_s_2_1[1],not_s_2_1[1]);
nand(s_0_en[1], not_s_0_en[1], not_s_0_en[1]);
nand(not_d[1], s_2_1[1], s_0_en[1]);

nand(not_s_2_1[2], not_s[2], s[1]);
nand(not_s_0_en[2], not_s[0], en);
nand(s_2_1[2], not_s_2_1[2],not_s_2_1[2]);
nand(s_0_en[2], not_s_0_en[2], not_s_0_en[2]);
nand(not_d[2], s_2_1[2], s_0_en[2]);

nand(not_s_2_1[3], not_s[2], s[1]);
nand(not_s_0_en[3], s[0], en);
nand(s_2_1[3], not_s_2_1[3],not_s_2_1[3]);
nand(s_0_en[3], not_s_0_en[3], not_s_0_en[3]);
nand(not_d[3], s_2_1[3], s_0_en[3]);

nand(not_s_2_1[4], s[2], not_s[1]);
nand(not_s_0_en[4], not_s[0], en);
nand(s_2_1[4], not_s_2_1[4],not_s_2_1[4]);
nand(s_0_en[4], not_s_0_en[4], not_s_0_en[4]);
nand(not_d[4], s_2_1[4], s_0_en[4]);
```

```
nand(not_s_2_1[5], s[2], not_s[1]);
nand(not_s_0_en[5], s[0], en);
nand(s_2_1[5], not_s_2_1[5],not_s_2_1[5]);
nand(s_0_en[5], not_s_0_en[5], not_s_0_en[5]);
nand(not_d[5], s_2_1[5], s_0_en[5]);

nand(not_s_2_1[6], s[2], s[1]);
nand(not_s_0_en[6], not_s[0], en);
nand(s_2_1[6], not_s_2_1[6],not_s_2_1[6]);
nand(s_0_en[6], not_s_0_en[6], not_s_0_en[6]);
nand(not_d[6], s_2_1[6], s_0_en[6]);

nand(not_s_2_1[7], s[2], s[1]);
nand(not_s_0_en[7], s[0], en);
nand(s_2_1[7], not_s_2_1[7],not_s_2_1[7]);
nand(s_0_en[7], not_s_0_en[7], not_s_0_en[7]);
nand(not_d[7], s_2_1[7], s_0_en[7]);

nand(d[0], not_d[0], not_d[0]);
nand(d[1], not_d[1], not_d[1]);
nand(d[2], not_d[2], not_d[2]);
nand(d[3], not_d[3], not_d[3]);
nand(d[4], not_d[4], not_d[4]);
nand(d[5], not_d[5], not_d[5]);
```

```
    nand(d[6], not_d[6], not_d[6]);

    nand(d[7], not_d[7], not_d[7]);


endmodule
```

Вход: 3-битная шина s с декодируемым числом и сигнал разрешения en.

Выход: 8-битная шина d, где порядок равного 1 бита соответствует декодированному числу.

Используются 6 вспомогательных (5 6-битных и 1 3-битная) шин для передачи результатов операции NAND.

not_s, not_d – инвертированные биты шин s и d.

(not_)s_2_1[n] - (не)инвертированные биты результата (not_)s[2] NAND (not_)s[1] (т.е. выполнение NAND для соответствующих числу n по инверсии старшего и среднего битов).

(not_)s_0_en[n] - (не)инвертированные биты результата (not_)s[0] NAND (not_)en (т.е. выполнение NAND для соответствующего числу n по инверсии младшего бита и сигнала разрешения).

Можно заметить, что в коде 8 раз повторяются разделённые переносом 5 строчек, которые отличаются только номером бита и присутствием/отсутствием приставки «not_» для нужной для этого числа инверсии.

Код разработанного тестового окружения БОЭ:

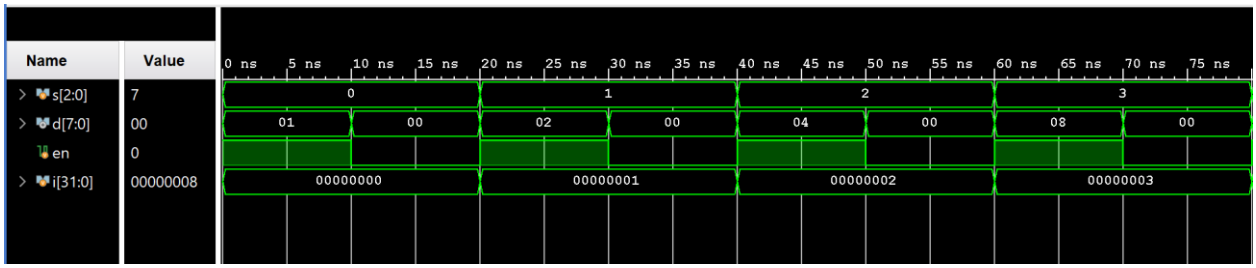
```
`timescale 1ns / 1ps


module decoder_tb;


    reg [2:0] s;
    wire[7:0] d;
    reg en;


    integer i;
```

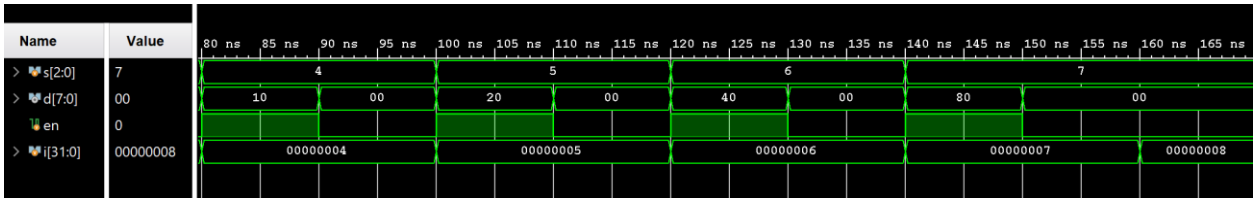
```
decoder decoder_1(
    .s(s),
    .d(d),
    .en(en)
);

initial begin
    for(i = 0; i < 8; i = i+1) begin
        s = i;
        en = 1;

        #10

        if (d == 2**i) begin
            $display("Correct! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
        end else begin
            $display("Incorrect! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
        end

        en = 0;

        #10

        if (d == 0) begin
            $display("Correct! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
        end else begin
            $display("Incorrect! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
        end
```

```
        end
        #10 $stop;


    end
endmodule
```

В тестовом окружении задаются переменные входа (s, en) и выхода (d), а также целочисленная переменная i, которая будет изменяться в цикле от 0 до 8; её значение будет присваиваться шине s. Затем проверяется работоспособность дешифратора при включенном и выключенном сигнале разрешения; так как число на входе соответствует порядку равного 1 бита, можно проверять это как равенство результата степени двойки этого числа.

Временная диаграмма процесса тестирования БОЭ:





Как можно видеть, во время процесса тестирования переменная i изменялась от 0 до 8 каждые 20 нс, переменная s – от 0 до 7 (в последний раз присваивания не происходило), сигнал en каждые 10 нс инвертировался, вследствие чего переменная d в течение этого промежутка времени менялась от степени двойки числа s к 0 и наоборот.

Вывод в консоль:

Correct! s=000, d=00000001, en=1, i=0

Correct! s=000, d=00000000, en=0, i=0

Correct! s=001, d=00000010, en=1, i=1

Correct! s=001, d=00000000, en=0, i=1

Correct! s=010, d=00000100, en=1, i=2

Correct! s=010, d=00000000, en=0, i=2

Correct! s=011, d=00001000, en=1, i=3

Correct! s=011, d=00000000, en=0, i=3

Correct! s=100, d=00010000, en=1, i=4

Correct! s=100, d=00000000, en=0, i=4

Correct! s=101, d=00100000, en=1, i=5

Correct! s=101, d=00000000, en=0, i=5

Correct! s=110, d=01000000, en=1, i=6

Correct! s=110, d=00000000, en=0, i=6

Correct! s=111, d=10000000, en=1, i=7

Correct! s=111, d=00000000, en=0, i=7

Выводы по работе:

В процессе выполнения данной работы я познакомился со средой Ltspice и языком описания аппаратуры Verilog. В качестве опытного образца я создал собственный вентиль NAND и на его основе создал и протестировал позиционный шифратор «3 в 8».